

IN THE CLAIMS:

1. (Currently Amended): A method for creating a software state machine, comprising:
providing a state machine object; and
providing an initializer object, wherein the initializer object defines states,
actions, and conditions for a software state machine,
wherein the state machine object is configured to use the initializer object to
create a table object,
wherein the state machine object is configured to create an array of state
variables,
wherein the table object is configured to create an array of state transition objects
based on the array of state variables and return the array of state transition objects to the
state machine object, and
wherein the state machine object is configured to execute the software state
machine using the array of state transition objects.
2. (Original): The method of claim 1, wherein the state machine object includes an object
constructor method.
3. (Original): The method of claim 2, wherein the object constructor method is
configured to create an instance of the initializer object.
4. (Canceled)
5. (Currently Amended): The method of claim [[4]] 1, wherein the initializer object
includes a table element array creation method and wherein the state machine object is
configured to call the table element array creation method to create the table object using
the results of the table element array creation method.
6. (Canceled)

7. (Currently Amended): The method of claim 1, wherein the initializer object includes a table variable array creation method and wherein the state machine object is configured to call the table variable array creation method to create the array of state variables using the results of the table variable array creation method.

8. (Canceled)

9. (Original): The method of claim 1, wherein at least one of the state machine object and the initializer object implements an interface.

10. (Original): The method of claim 1, wherein the state machine object includes a state method that is configured to return a current state of the software state machine.

11. (Currently Amended): A method for creating software state machines, comprising:

providing a state machine object;

~~creating a first instance of the state machine object with a first state machine initializer, wherein the first instance of the state machine object executes a first software state machine; and~~

~~creating a second instance of the state machine object with a second state machine initializer, wherein the second instance of the state machine object executes a second software state machine~~ providing an initializer object, wherein the initializer object defines a plurality of states, one or more actions, one or more inputs, one or more conditions, one or more events, and one or more triggers for a software state machine, wherein the state machine object is configured to use the initializer object to create an array of state transition objects, wherein each state transition object in the array of state transition objects defines at least one of the one or more conditions that causes a given state transition in the software state machine, and wherein each condition is a Boolean expression formed from at least one of the one or more inputs; and

responsive to occurrence of a trigger, evaluating the one or more inputs, computing the one or more conditions, and determining a next state based on the array of state transition objects.

12. (Currently Amended): An apparatus for creating a software state machine, comprising:

a processor; and

a memory having stored therein a state machine object and an initializer object, wherein the initializer object defines states, actions, and conditions for a state

machine,

wherein the state machine object is configured to use the initializer object to create a table object,

wherein the state machine object is configured to create an array of state variables,

wherein the table object is configured to create an array of state transition objects based on the array of state variables and return the array of state transition objects to the state machine object, and

wherein the state machine object is configured to execute the software state machine using the array of state transition objects.

13. (Original): The apparatus of claim 12, wherein the state machine object includes an object constructor method.

14. (Original): The apparatus of claim 13, wherein the object constructor method is configured to create an instance of the initializer object.

15. (Canceled)

16. (Currently Amended): The apparatus of claim ~~[[15]]~~ 12, wherein the initializer object includes a table element array creation method and wherein the state machine object is configured to call the table element array creation method to create the table object using the results of the table element array creation method.

17. (Canceled)

18. (Currently Amended): The apparatus of claim 12, wherein the initializer object includes a table variable array creation method and wherein the state machine object is configured to call the table variable array creation method to create an array of state variables using the results of the variable array creation method.

19. (Canceled)

20. (Original): The apparatus of claim 12, wherein at least one of the state machine object and the initializer object implements an interface.

21. (Original): The apparatus of claim 12, wherein the state machine object includes a state method that is configured to return a current state of the software state machine.

22. (Original): An apparatus, comprising:

- a state machine initializer object;

- a state machine object; and

- a virtual machine, wherein the virtual machine creates an instance of the state machine object including an object constructor method,

- wherein the object constructor method creates an instance of the initializer object and uses the instance of the initializer object to create a table object and an array of state variables,

- wherein the table object includes a state array creation method and the object constructor method calls the state array creation method to create an array of state transition object, and

- wherein the instance of the state machine object uses the array of state transition objects to execute a software state machine.

23. (Currently Amended): A computer program product, in a computer readable medium, for creating a software state machine, comprising:

- instructions for providing a state machine object; and

instructions for providing an initializer object, wherein the initializer object defines states, actions, and conditions for a software state machine,

wherein the state machine object is configured to use the initializer object to create a table object,

wherein the table object is configured to create an array of state variables,

wherein the table object is configured to create an array of state transition objects based on the array of state variables and return the array of state transition objects to the state machine object,

wherein the state machine object is configured to receive the array of state transition objects[[,]] and execute the software state machine using the array of state transition objects.

24. (Currently Amended): A computer program product, in a computer readable medium, for creating software state machines, comprising:

~~instructions for creating a first instance of a state machine object with a first state machine initializer, wherein the first instance of the state machine object executes a first software state machine providing a state machine object; [[and]]~~

~~instructions for creating a second instance of the state machine object with a second state machine initializer, wherein the second instance of the state machine object executes a second software state machine providing an initializer object, wherein the initializer object defines a plurality of states, one or more actions, one or more inputs, one or more conditions, one or more events, and one or more triggers for a software state machine, wherein the state machine object is configured to use the initializer object to create an array of state transition objects, wherein each state transition object in the array of state transition objects defines at least one of the one or more conditions that causes a given state transition in the software state machine, and wherein each condition is a Boolean expression formed from at least one of the one or more inputs; and~~

~~instructions, responsive to occurrence of a trigger, for evaluating the one or more inputs, computing the one or more conditions, and determining a next state based on the array of state transition objects.~~

25. (New): The method of claim 11, wherein a given state transition object in the array of state transition objects defines an action to take responsive to a given state transition.

26. (New): The method of claim 11, wherein a given state transition object in the array of state transition objects defines an event to be generated responsive to a given state transition.

27. (New): The method of claim 11, wherein providing an initializer object includes:
presenting one or more graphical user interfaces for defining the plurality of states, the one or more actions, the one or more inputs, the one or more conditions, the one or more events, and the one or more triggers for the software state machine.

28. (New): The computer program product of claim 24, wherein a given state transition object in the array of state transition objects defines an action to take responsive to a given state transition.

29. (New): The computer program product of claim 24, wherein a given state transition object in the array of state transition objects defines an event to be generated responsive to a given state transition.

30. (New): The computer program product of claim 24, wherein the instructions for providing an initializer object includes:

instructions for presenting one or more graphical user interfaces for defining the plurality of states, the one or more actions, the one or more inputs, the one or more conditions, the one or more events, and the one or more triggers for the software state machine.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.